LIMITATIONS OF CHALLENGE-RESPONSE ENTITY AUTHENTICATION

Chris Mitchell

Hewlett-Packard Laboratories
Filton Road
Stoke Gifford
Bristol BS12 6QZ
England

Version 2.0 - 16th May 1989

Abstract

We consider two basic versions of the challenge-response authentication protocol, and exhibit both a method of attack and a simple modification preventing such attacks. We go on to consider three variants of the basic protocols, and show that one of them is completely insecure.

Introduction

Challenge-response protocols are widely used for identity verification over insecure channels. Two versions of the challenge-response identification protocol are described in Davies and Price's book, [3]. The simplest, described on p.185, we call *Protocol P*, and is as follows. Suppose that A and B are two entities wishing to verify the identity of each other. Further suppose that A and B share a piece of secret information K (known as a *password* or *key*), which they will use to check each other's identity. Finally, also suppose that O is a *one-way function*, i.e. a function that is easy to compute yet very difficult to invert. A and B then exchange messages P1-P3 (where A -> B indicates a message sent from A to B, and X,Y indicates the concatenation of items X and Y):

P1: A -> B: R_A (a random value)

P2: B -> A: R_B, O(K, R_A) (R_B is another random value)

P3: A -> B: O(K, R_B)

A knows that B sent P2 since, apart from A, only B knows K, and, similarly, B knows that A sent P3. The use of the random values R_A and R_B prevents the replay of previously valid exchanges.

Such schemes are widely used. In particular, *dynamic password* schemes use part of Protocol P; they usually omit R_B from message P2 and the whole of P3. Dynamic password schemes are used in the following situation.

Suppose B is a remote user of a computer system A, and that B has asked to *log on* to A. In order to check B's claimed identity, A sends B a *challenge* (R_A) and B returns O(K, R_A). The part of the protocol not used would enable B to verify the identity of A; this is normally assumed to be unnecessary for remote login, although this could be a useful service. Dynamic password systems have been readily available for some time, typically involving users being supplied with calculator-like devices equipped with a secret key known only to the central computer. Human users are required to copy the challenge (R_A) from their terminal into the device, and then copy back the response, i.e. O(K, R_A). For further details about such devices see, for example, Beker, [2].

Because of the importance of the basic protocol, there have been recent moves within I.S.O. to standardise one version of it within a larger standard covering four different peer-entity authentication techniques. A draft document was submitted for voting as an I.S.O. Draft proposal during 1988,

[5]. The version described in that document differs to some extent from the version described above, although the basic idea remains the same. The three-message protocol (*Protocol Q*) is as follows (where E represents a (reversible) encryption function, and E_K represents the particular encryption function prescribed by key K).

Q1: A -> B: R_A (a random value)

Q2: B -> A: $E_K(R_A, R_B)$ (R_B is another random value)

Q3: A -> B: R_B

A weakness in the authentication process

We now show that both protocols P and Q are subject to possible *reflection* attacks. Suppose A and B authenticate each other using Protocol P, and that user C wishes to impersonate B to A. The attack essentially involves running two copies of the protocol simultaneously, as we now see.

A initiates the protocol by sending P1, i.e. A sends R_A to C. Having received R_A, C inaugurates a second copy of the same protocol but in the reverse direction, by sending R_A to A and pretending that it has come from B. A responds to the receipt of this message by sending back $O(K, R_A)$ concatenated with another random value, R_A' say. C , having received this message, uses it to respond to A's initial challenge, thereby successfully completing the authentication protocol. A believes itself to be talking to B when, in fact, B has not

been involved in any of the communications. The same type of attack will also work with Protocol Q.

It might be argued that the above scenario is unrealistic, and that in most circumstances A will not permit two versions of the protocol to run simultaneously. However, this protocol might well be used in a networked computing environment, where each computer is capable of simultaneously communicating with a number of other computers over a number of channels. In this case, the above attack would probably be easy to operate.

One possible fix would be to insist that, whenever this protocol is used, checks are always included to prevent two copies of it running simultaneously. However, this is both inelegant and unnecessarily restrictive, since other, simpler means of avoiding the problem are available.

Basically, the reflection attack works because of symmetry in the protocol. Messages P2 and Q2 have the same form regardless of whether they are passed from A to B or vice versa. Two possible solutions arise immediately from this observation.

The first involves A and B sharing two secret keys, KAB and KBA, where KAB is only ever used to prepare messages by A and is used by B for checking messages from A (and vice versa). With this modification, P2, P3 and Q2 become:

P2': B -> A: R_B, O(KBA, R_A)

P3': A -> B: O(KAB, R_B)

Q2': B -> A: E_{KBA}(R_A, R_B)

To be fair, Davies and Price essentially suggest this protocol on p.185 of their book, [3], but without stating why. The second possible solution involves inserting the name of the message originator in P2, P3 and Q2 as follows, thereby preventing their re-use in the reverse direction:

P2": B -> A: R_B, O(K, R_A, B)

P3": A -> B: O(K, R_B, A)

Q2": B -> A: E_K(R_A, R_B, B)

Both of these solutions require very little extra overhead.

Extensions to the protocol and their limitations

Because of the usefulness and simplicity of Protocols P and Q, a number of variations have been introduced in order to provide a wider range of security services. We consider three of them here, and, where relevant, note their limitations.

We start by considering the second protocol given in Davies and Price's book, [2]. On pages 140,141 they suggest the use of the following variant of Protocol P, which we call Protocol R. This variant simultaneously provides message integrity and

message origin authentication for the three messages M_1, M_2 and M_3.

R1: A -> B: R_A, M_1, O(K, M_1, R_A)

R2: B -> A: R_B, M_2, O(K, M_2, R_A, R_B)

R3: A -> B: M_3, O(K, M_3, R_B)

This protocol seems sound, although, as Davies and Price point out, B cannot trust the validity of message M_1 until the receipt of R3. However, the protocol is still potentially subject to reflection attacks. Introducing some asymmetry, as described above, would prevent these attacks. Note that the inclusion of R_B within the scope of O in R2 is not essential to the protocol.

It is interesting to note that a version of Protocol R has been proposed for use with a Dynamic Password scheme (see Beker, [2]). In this latter case M_1 and O(K, M_1, R_A) are omitted from R1, the value R_B is omitted from R2, and R3 is omitted altogether.

The second variation we consider is described by Diffie, [4]. This protocol, a variation of Protocol Q we call Protocol S, provides key exchange for connection security.

S1: A -> B: E_K(SK), E_{SK}(R_A)

S2: B -> A: E_{SK}(R_A, R_B)

S3: A -> B: E_{SK}(R_B)

SK is a *session key* used only for the duration of one protocol exchange. Diffie, [4], goes on to describe how R_A and R_B, since they are secret, can be used to provide a range of security services for subsequent communications between A and B.

This protocol appears fundamentally sound. However, as before, it should be clear that this protocol is potentially subject to reflection attacks. This is not a problem for the particular situation in which Diffie proposes to use the protocol, although it should be noted by anyone wishing to use this protocol in other environments. As before, the problem can be rectified by the addition of asymmetry into the protocol.

The third protocol variant we consider is due to Arditti et al., [1]. They propose using a modified version of Protocol P to directly provide authentication and/or encryption of short messages. To provide just message authentication they use a simplified version of Protocol R. To provide message encryption they use protocol T, as below:

T1: A -> B: R_A

T2: B -> A: M + O(K, R_A)

where, for example, O(K, R_A) and M are made up of 8-bit characters and K+M denotes character-wise addition modulo 256. Of course M may only have at most as many characters as O(K, R_A).

Suppose C observes such a protocol exchange and wishes to discover M. C now impersonates A and sends R_A to B (that is the same value of R_A as used in the observed protocol). If B has a new message M' to send to A then B will respond with

$$M' + O(K, R_A).$$

C will then be able to immediately compute the value M + M', which may be sufficient to reveal part of the contents of both M and M' (if M and M' contain sufficient redundancy). Moreover M' may actually be empty, in which case C can immediately deduce M. In any case the protocol is insecure and should never be used. Arditti et al. go on to propose a similar protocol attempting to provide both confidentiality and authentication for a short message - this protocol is also subject to the same kind of attack.

Summary

From the above discussion it should be clear that the challenge-response protocol is basically sound if used with care. In general, because of the possibility of reflection attacks, it would seem a wise precaution to always use an asymmetric form of the protocol, using one of the two possible modifications suggested above. The protocol can be used for peer-entity authentication and key-exchange in addition to user identification. However any such extensions should be carefully analysed before use; otherwise problems of the type encountered in Protocol T above may arise.

References

[1] ARDITTI, D., CAMPANA, M., ALLEGRE, F., and COHEN, R.: 'Access control with a minimal investment'. Proceedings of SECURICOM 89, Paris, 1989, pp.159-169.

[2] BEKER, H.J.: 'Secure access control to host applications'. Proceedings of the Second IEE International Conference on Secure Communications, London, 1986, pp.20-21.

[3] DAVIES, D.W., and PRICE, W.L.: 'Security for computer networks' (John Wiley, Chichester, 1984).

[4] DIFFIE, W.: 'Protection of terminal to host communications by end-to-end cryptography'. Proceedings of SECURICOM 89, Paris, 1989, pp.331-338.

[5] I.S.O.: 'Peer entity authentication mechanisms using an N-bit secret-key algorithm', ISO/IEC/JTC1/SC20/WG1/N153, April 1988.